

Distributed Systems

COMP90015 2019 Semester 1
Tutorial 10

Questions for today

- What is a domain transition and how many domain transitions are required for an RPC/RMI call from one process to another on the same host?
- What is process migration and explain two circumstances that can prohibit or make it difficult for process migration to take place?
- Why is shared memory faster than message passing when communicating between processes on the same host? State a reason why shared memory is not preferable when communicating between processes on the same host.
- Give two reasons why a monolithic kernel is sometimes more efficient than a micro kernel. Give three reasons why a micro kernel is preferable to a monolithic kernel.
- What are some key design issues for distributed file systems?

- What is a domain transition and how many domain transitions are required for an RPC/RMI call from one process to another on the same host?

- What is a domain transition and how many domain transitions are required for an RPC/RMI call from one process to another on the same host?

Domain Transition happens when the protection mode is changed, that is, when there is a change from a user context to a kernel context.

As this call is on the same host, the domain transition is to the kernel, to the other process, back to the kernel and back to the calling process.

→ 4 domain transitions.

- What is process migration and explain two circumstances that can prohibit or make it difficult for process migration to take place?

- What is process migration and explain two circumstances that can prohibit or make it difficult for process migration to take place?

Process Migration is a form of process management where the process is moved from one host to another.

This can be done by copying their address space.

It can be difficult if:

- The process code is CPU dependent.
- The process is using host resources such as open files and sockets.

- Why is shared memory faster than message passing when communicating between processes on the same host? State a reason why shared memory is not preferable when communicating between processes on the same host.

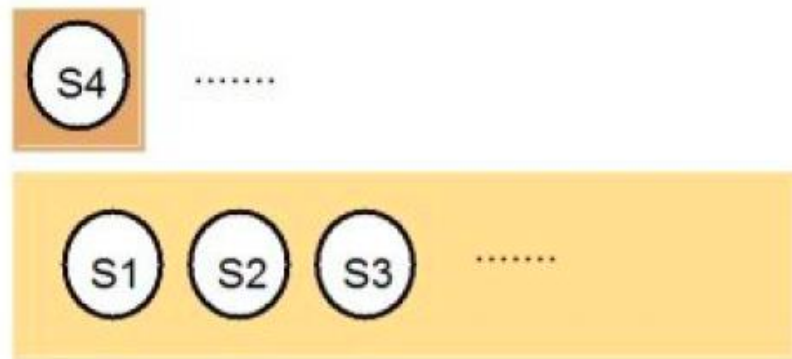
- Why is shared memory faster than message passing when communicating between processes on the same host? State a reason why shared memory is not preferable when communicating between processes on the same host.

When we use message passing, there is an overhead of marshaling, transmission and unmarshalling the messages. There is no such overhead in a shared memory.

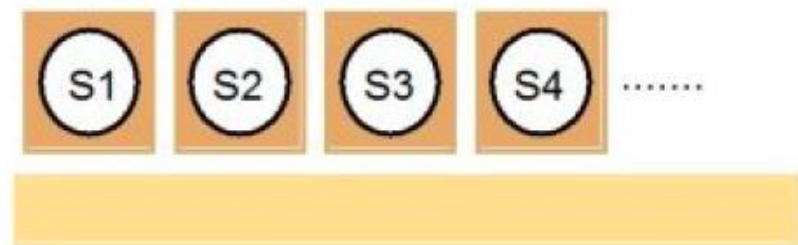
Disadvantage:

When using shared memory processes are not protected from each other, if we want more security and protection we should use message passing.

- Give two reasons why a monolithic kernel is sometimes more efficient than a micro kernel. Give three reasons why a micro kernel is preferable to a monolithic kernel.



Monolithic Kernel



Microkernel

Key:

Server: ○ Kernel code and data: ■

Dynamically loaded server program: ■

- Give two reasons why a monolithic kernel is sometimes more efficient than a micro kernel. Give three reasons why a micro kernel is preferable to a monolithic kernel.

Monolithic Kernel provides diverse functionality.

Complex functionality can be provided through a smaller number of system calls and interprocess communication is minimized.

Micro Kernel provides only fundamental functionality.

Allow a greater extensibility and have a greater ability to enforce modularity behind memory protection boundaries.

A small kernel is also more likely to be free of bugs.

- What are some key design issues for distributed file systems?

- What are some key design issues for distributed file systems?
 - Effective use of client caching to achieve performance
 - Maintain consistency between multiple copies of cached files
 - Recovery after client or server failure
 - High throughput for reading and writing files of all sizes
 - Scalability